Trees

- regression trees
- classification trees

Regression trees

IDEA: approximate a function $f(x)$ with a step function (function piecewise constant)

We need to decide:

(a) how many splits in the x-axis (when $p=1$)

(b) where to place these splits

(c) which value to use to approximate $f(x)$

The last point (c) is the simplest:

$$\frac{1}{|R_h|}\int_{R_h} f(x)\,dx$$

R_h is the interval

$|R_h|$ is the lenght ($p=1$) of the interval

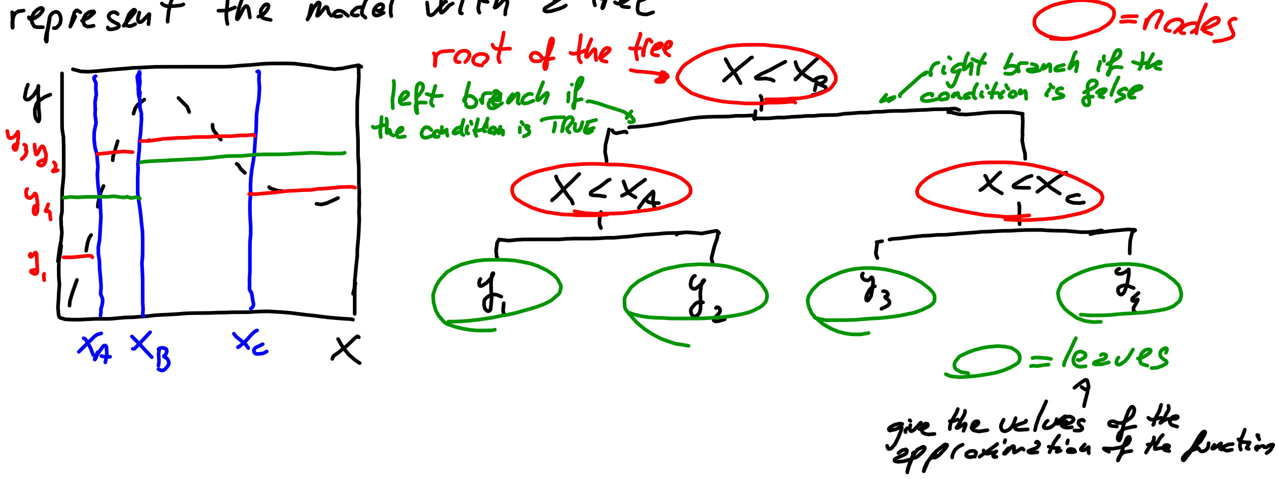
→ when we have data $\hat{y} = \frac{1}{\# x_i \in R_j} \sum_{i: x_i \in R_j} y_i$

About point (b), the intuitive choice is to have more intervals where the function is steepest (differences are bigger)

The number of split (a) is related to the concept of model complexity
- more splits, better approximation, but more complex model (in the case of prediction, may lead to overfit)

This approximation scale very easily to higher dimensions (e.g. columns of X)
With $p=2$, R_h are rectangulars, but the idea is the same

The nice part of this way of approximating $f(x)$ is that we can represent the model with a tree



Graphical tree representation is not as attractive as the graphic in figure 8.2 but
- it is easier to store (only needs nodes and leaves);
- it can be improved really simply, by adding a new node where a leaf is
  ↳ recursive improvement which increases accuracy
  ↓
  grow of a tree

Grow the tree

when $p=1$, intervals
in the general sense also for $p>2$

The goal is to estimate $f(x)$ in a form

$$\hat{f}(x) = \sum_{h=1}^{J} c_h \mathbb{1}(x \in R_h)$$

where c_h, $h=1,\dots,J$ are constants and R_h, $h=1,\dots,J$ are the rectangular

Obviously, we want to minimize the deviance
→ globally (only one step) is computationally infeasible
→ step-by-step procedure (starting from one node (root), we improve the approximation step by step using the previous approximation)

Let us start with the formulation of the deviance

$$D = \sum_{i=1}^{n} (y_i - f(x))^2 = \sum_{h=1}^{J} \left[\sum_{i:x_i \in R_h} (y_i - c_h)^2 \right] = \sum_{h=1}^{J} D_h$$

→ can be done due to the division of the support of X in non-overlapping rectangulars

Remember that $\underset{a}{\operatorname{argmin}} \sum_{i=1}^{n} (z_i - a)^2 = \bar{z}$ (average)

For each dimension of X, x_j, $j=1,\dots,p$
for each R_h, $h=1,\dots,J$
find the best split R_h in R_h' and R_h'' such that $g_h = D_h - D_h^*$ is maximum, where

$$D_h = \sum_{i:x_i \in R_h} (y_i - c_h)^2$$

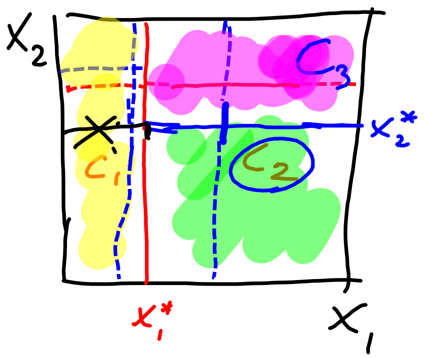
where c_h is the average of y_i, $i:x_i \in R_h$

$$D_h^* = \sum_{i:x_i \in R_h'} (y_i - c_h')^2 + \sum_{i:x_i \in R_h''} (y_i - c_h'')^2$$

c_h' is the average related to R_h'
c_h'' is the average " " R_h''

choose the best j and the best h (best = larger g_h)

→ repeat the procedure for a large number of times (potentially until R_h contain only one observation, $\forall h$)



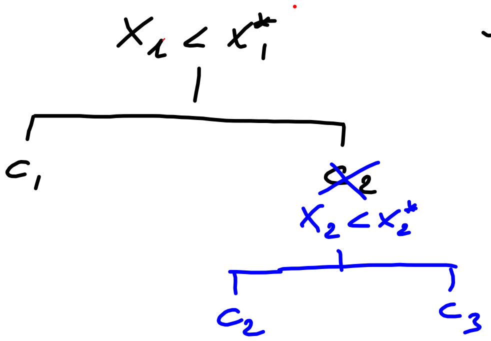
1^{st} step (I choose to split $h=1$ and $p=1$)

2^{nd} step (I choose to split $h=2$ - related to c_2 - and $x=2$)

→ note that at each step we divide only <u>one</u> rectangular in two parts

→ it does not guarantee to find a global minimum;

→ it is an acceptable solution in a feasible computational complexity



The procedure will lead to a continuous decrease of the deviance until each leaf contains only one observation (which corresponds to interpolate)

→ we do not want that (overfitting)

→ we need to <u>prune</u> the tree!

Prune the tree

Minimization of the deviance → overfitting

SOLUTION: add a penalty term to penalize complexity (# leaves)

$$C_\alpha(J) = \sum_{h=1}^{J} D_h + \alpha J$$

where α is our "usual" tuning parameter.

→ it can be shown that, for each α, there is a unique smallest tree which minimizes $C_\alpha(J)$, see Breiman et al (1984, prop 3.7)

⇒ we only need to find the best α, then we can find the best tree

CROSS-VALIDATION

(minimize the cross-validated deviance)

↳ deviance computed on the k-th fold using a tree trained on the other k-1 folds

remove sequentially a leaf, each time that that leads to the smallest increase in the deviance until $C_\alpha(J)$ starts to increase

Prediction of a new observation x_0

- starting from the root, we follow the path based on the nodes until we reach a leaf (the value in the leaf is $\hat{y}_0$)

E.g.: $\hat{f}(x_0 = 1.2) = 0.5395$

$x_0 > 0.54$ → right
$x_0 < 1.52$ → left
$x_0 > 0.71$ → right → find leaf → 0.5395

About variables:

- usually pruning leads to small trees => only a few variables are used in the nodes => a lot of variables (dimensions of x) are not used → these variables are not relevant (automatic variable selection)
- among those that are selected, it is hard to say which are the most important
  - the gain in deviance at each node is related only to a specific dichotomization, not to the whole variable;
  - but for the root, the dichotomization is only related to a specific rectangular;
  - several nodes use the same variables.

Attempts to construct measures of variable importance based on the squares of the gains S_h